

## Nogle kommandoer til brug for en Dijkstra tegneserie

Vi skal bruge pakken **GraphTheory**. Heri findes kommandoerne **Graph**, **HighlightVertex**, **DrawGraph**, **InducedSubgraph**, **HighlightTrail**, **DijkstrasAlgorithm**

```
> restart; with(GraphTheory):
```

En vægtet graf fastlægges

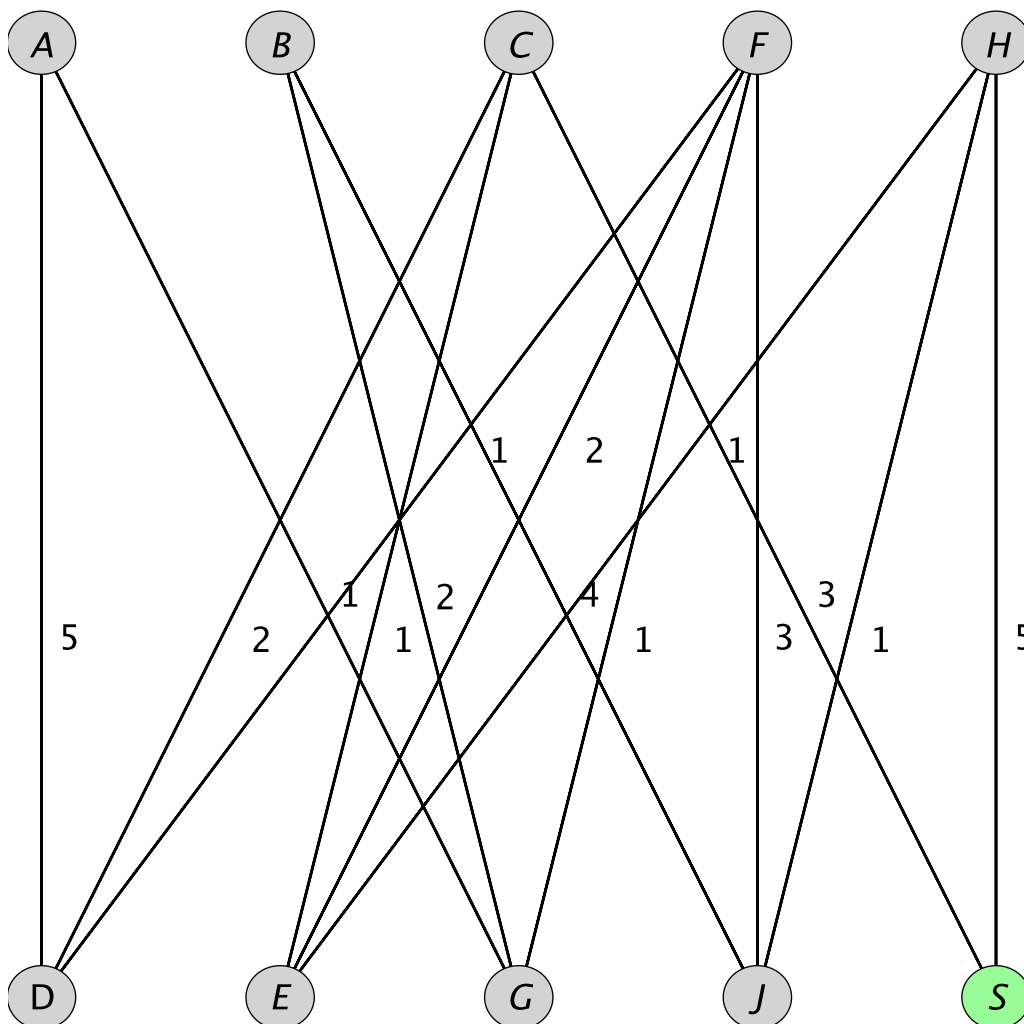
```
> GR:=Graph([ [{S,C},3], [{C,D},2], [{D,A},5], [{A,G},1],  
  , [{G,F},1], [{F,D},1], [{F,E},2], [{E,C},1], [{E,H},1],  
  , [{H,S},5], [{H,J},1], [{J,F},3], [{J,B},4], [{B,G},2]  
  ] );
```

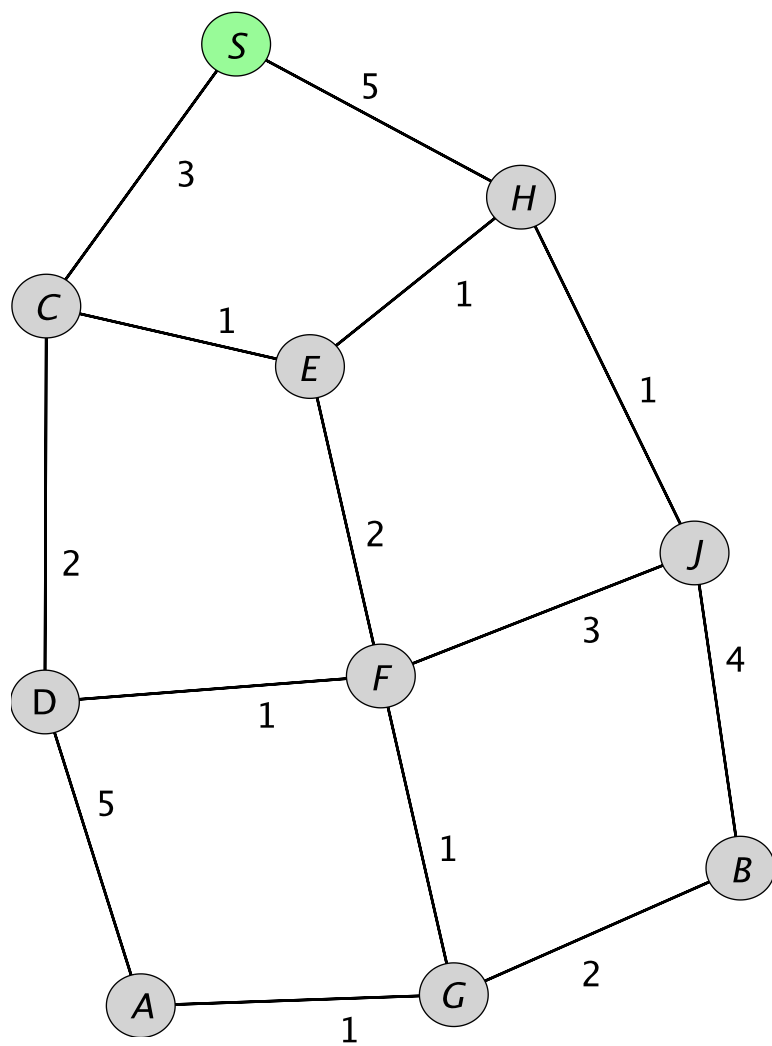
*GR := Graph 1: an undirected weighted graph with 10 vertices and 14 edge(s)*

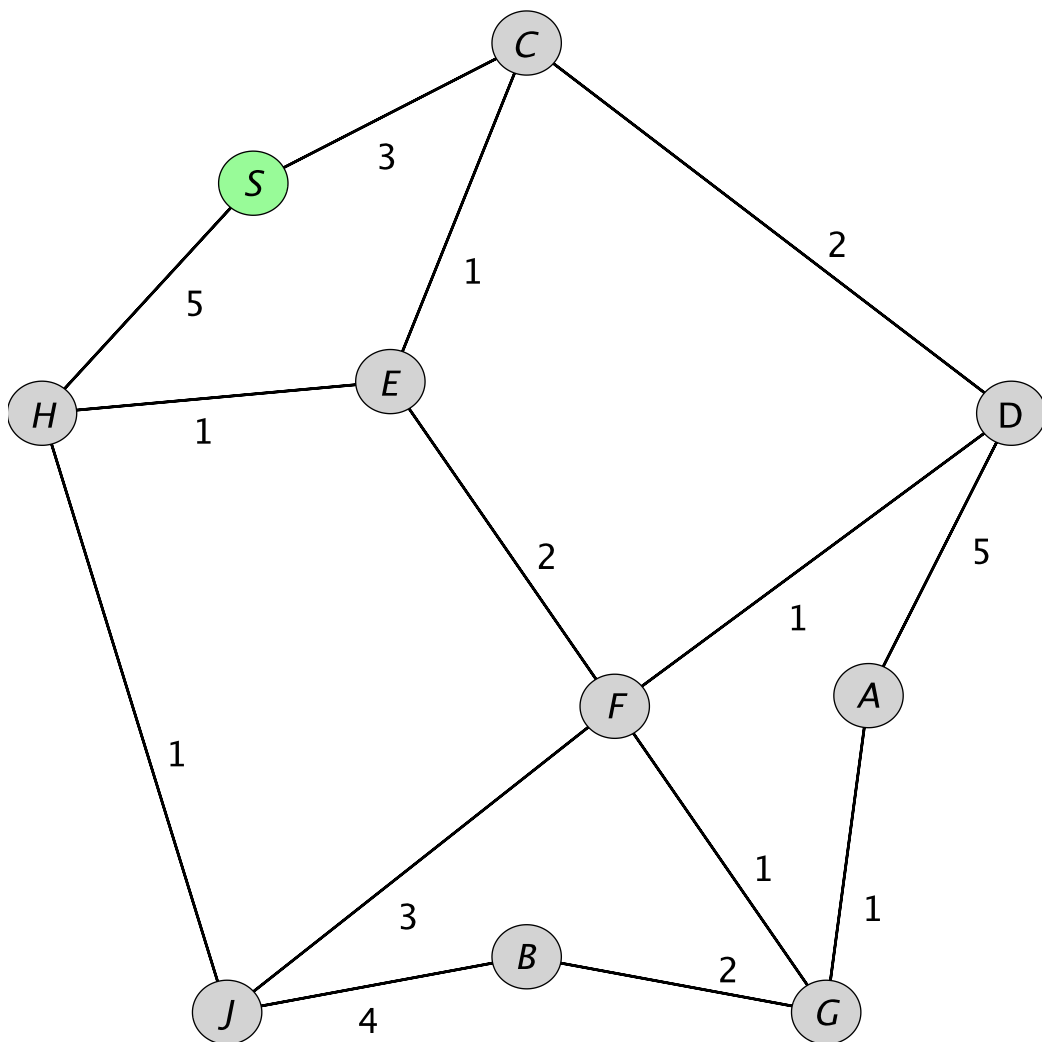
(1)

og tegnes - der er forskellige stilarter for det (NB! Det kan ske at nogle stilarter ikke kan lade sig gøre af grafteoretiske årsager). Vi fremhæver hjørnet S.

```
> HighlightVertex(GR,S); HighlightVertex(GR); DrawGraph(GR,style=  
  spring); DrawGraph(GR,style=planar);
```







Vi vælger fra nu **style=planar**. Vi finder *delgraften fastlagt ved kanter til S* og beregner korteste afstande. Til at beregne korteste afstande bruger Maple Dijkstras Algoritme på delgraferne  $S_1, S_2$ , osv., men I er i stand til at kontrollere Maple's beregninger

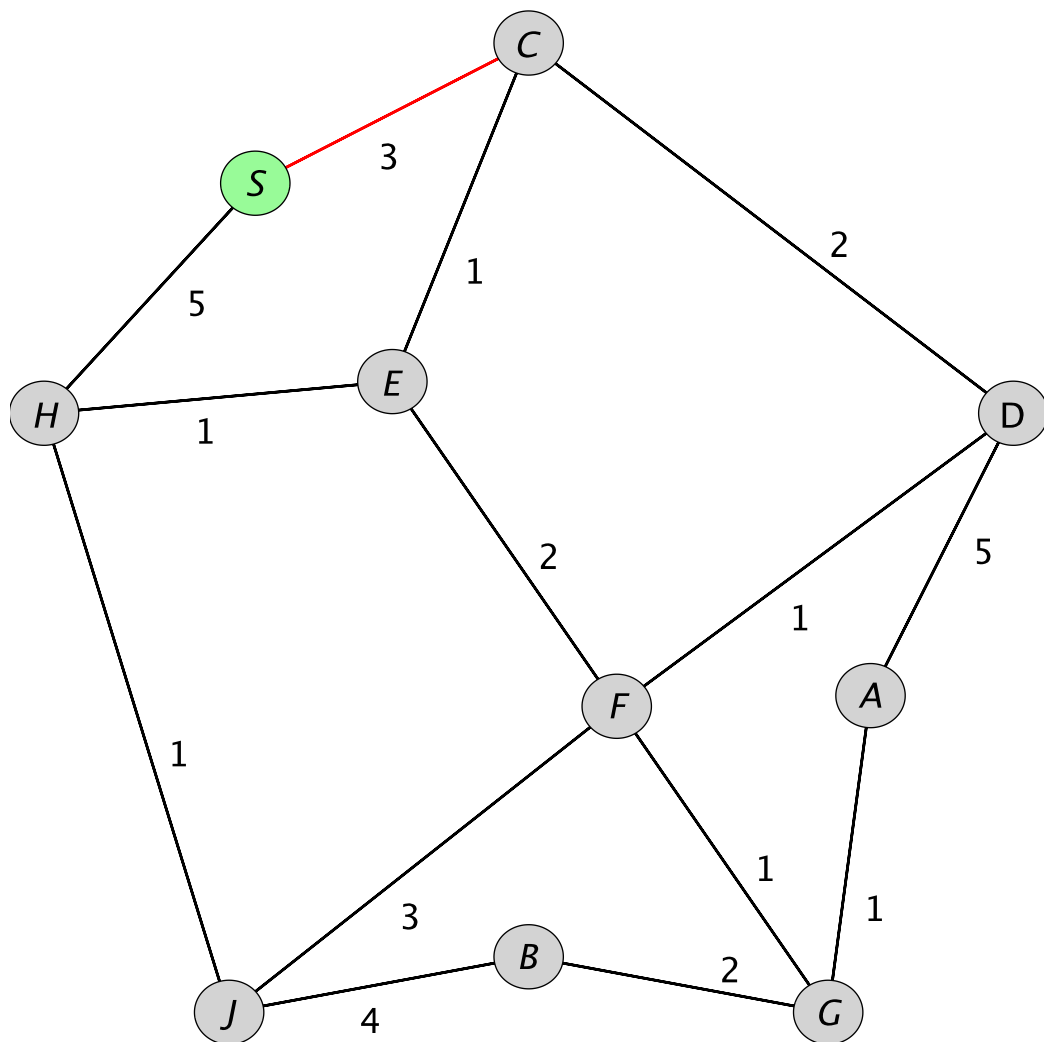
```
> S[1]:=InducedSubgraph(GR, [S,H,C]);DijkstrasAlgorithm(S[1],S);
      S1 := Graph 2: an undirected weighted graph with 3 vertices and 2 edge(s)
```

```
[[[S], 0], [[S, H], 5], [[S, C], 3]]
```

(2)

og tager *stien til det nærmeste hjørne, C*, nummer 3 i output (2)

```
> HighlightTrail(GR,(2)[3][1]);DrawGraph(GR,style=planar);
```



Dette var første trin i Dijkstras algoritme. Vi finder *delgrafen fastlagt ved hjørner som er kantforbundne til hjørner vi allerede har bestemt afstande til*.

```
> S[2] := InducedSubgraph(GR, [S,H,C,E,D]); DijkstrasAlgorithm(S[2], S);
```

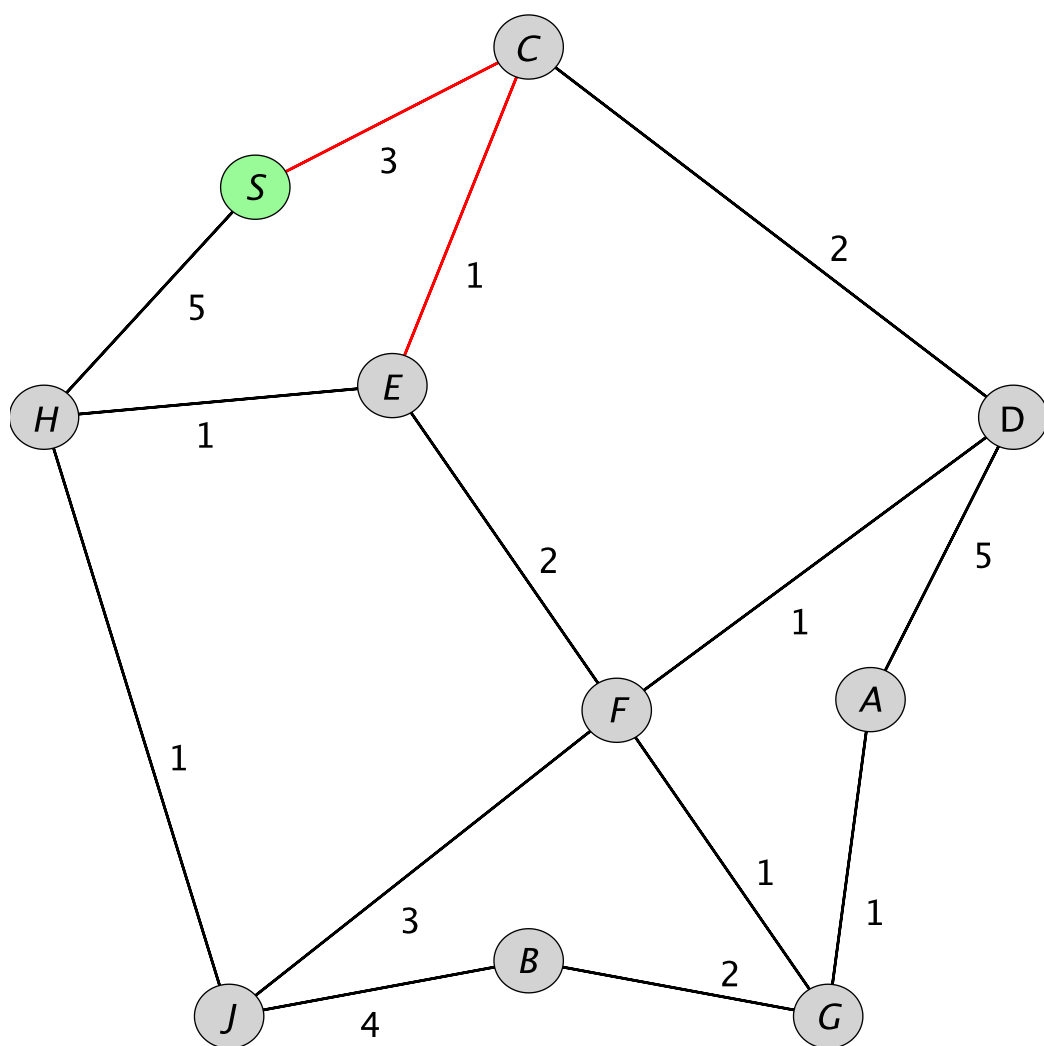
$S_2$  := Graph 3: an undirected weighted graph with 5 vertices and 5 edge(s)

```
[[[S], 0], [[S, H], 5], [[S, C], 3], [[S, C, E], 4], [[S, C, D], 5]]
```

(3)

og bestemmer *det næste nærmeste hjørne*, E, nummer 4 i output (3).

```
> HighlightTrail(GR, (3)[4][1]); DrawGraph(GR, style=planar);
```



Fortsæt selv til I har nået alle hjørner.